data that defines the 3D image displayed within the foregoing X window. Similarly, slave pipelines 58 and 59 render graphical data to frame buffers 68 and 69, respectively, via the X server 202 and the OGL daemon 205 within the pipelines 58 and 59.

Note that the graphical data rendered by each pipeline 56-59 defines a portion of the overall image to be displayed within region 249. Thus, it is not necessary for each pipeline 56-59 to render all of the graphical data defining the entire 3D image to be displayed in region 249. Indeed, in the preferred embodiment, each slave pipeline 56-59 preferably discards the graphical data that defines a portion of the image that is outside of the pipeline's responsibility. In this regard, each pipeline 56-59 receives from master pipeline 55 the graphical data that defines the 3D image to be displayed in region 249. Each pipeline 56-59, based on the aforementioned inputs received from slave controller 261 then determines which portion of this graphical data is within pipeline's responsibility and discards the graphical data outside of this portion.

For example, as described previously, slave pipeline 56 is responsible for rendering the graphical data defining the image to be displayed within portion 266 of FIG. 8. This portion 266 includes graphical data associated with screen relative coordinates (700, 1000) to (1000, 1300). Thus, any graphical data having screen relative coordinates outside of this range is discarded by the pipeline 56, and only graphical data having screen relative coordinates within the foregoing range is rendered to frame buffer 66.

Furthermore, slave pipeline 57 is responsible for rendering the graphical data defining the image to be displayed within portion 267 of FIG. 8. This portion 267 includes graphical data associated with screen relative coordinates (1000, 1000) to (1300, 1300). Thus, any graphical data having screen relative coordinates outside of this range is discarded by the pipeline 57, and only graphical data having screen relative coordinates within the foregoing range is rendered to frame buffer 67.

In addition, slave pipeline 58 is responsible for rendering the graphical data defining the image to be displayed within portion 268 of FIG. 8. This portion 268 includes graphical data associated with screen relative coordinates (700, 700) to (1000, 1000). Thus, any graphical data having screen relative coordinates outside of this range is discarded by the pipeline 58, and only graphical data having screen relative coordinates within the foregoing range is rendered to frame buffer 68.

Also, slave pipeline 59 is responsible for rendering the graphical data defining the image to be displayed within portion 269 of FIG. 8. This portion 269 includes graphical data associated with screen relative coordinates (1000, 700) to (1300, 1000). Thus, any graphical data having screen relative coordinates outside of this range is discarded by the pipeline 59, and only graphical data having screen relative coordinates within the foregoing range is rendered to frame buffer 69.

To increase the efficiency of the system 50, each slave pipeline 56-59 preferably discards the graphical data outside of the pipeline's responsibility before significantly processing any of the data to be discarded. Bounding box techniques may be employed to enable each pipeline 56-59 to quickly discard a large amount of graphical data outside of the pipeline's responsibility before significantly processing such graphical data.

In this regard, each set of graphical data transmitted to pipelines 56-59 may be associated with a particular set of bounding box data. The bounding box data defines a graphical bounding box that contains at least each pixel included in the graphical data that is associated with the bounding box data. The bounding box data can be quickly processed and analyzed to determine whether a pipeline 56-59 is responsible for rendering any of the pixels included within the bounding box. If the pipeline 56-59 is responsible for rendering any of the pixels included within the bounding box, then the pipeline 56-59 renders the received graphical data that is associated with the bounding box. However, if the pipeline

56-59 is not responsible for rendering any of the pixels included within the bounding box, then the pipeline 56-59 discards the received graphical data that is associated with the bounding box, and the pipeline 56-59 does not attempt to render the discarded graphical data. Thus, processing power is not wasted in rendering any graphical data that defines an object outside of the pipeline's responsibility and that can be discarded via the utilization of bounding box techniques as described above. Bounding box techniques are more fully described in U.S. Patent Number 5,757,321, entitled "Apparatus and Method for Clipping Primitives Using Information from a Previous Bounding Box Process," which is incorporated herein by reference.

After the pipelines 56-59 have respectively rendered graphical data to frame buffers 65-69, the graphical data is read out of frame buffers 65-69 through conventional techniques and transmitted to compositor 76. Through techniques described in more detail hereafter, the compositor 76 is designed to composite or combine the data streams from frame buffers 65-69 into a single data stream and to render the data from this single data stream to display device 83.

Once the graphical data produced by the application 17 has been rendered to display device 83, as described above, the display device 83 should display an image defined by the foregoing graphical data. This image may be modified by rendering new graphical data from the application 17 via the same techniques described hereinabove. For example, assume that it is desirable to display a new 3D object 284 on the screen 247, as shown by FIG. 10. In this example, assume that an upper half of the object 284 is to be displayed in the portion 266 and that a bottom half of the object is to be displayed in the portion 268. Thus, the object is not to be displayed in portions 267 and 269.

In the foregoing example, graphical data defining the object 284 is transmitted from client 52 to master pipeline 55. The master pipeline 55 transmits this graphical data to each